Optimizing Protein Sequence Design Based on the Advanced ProteinMPNN Deep Learning Model

Chenjie Li 36920241153226
Deyi Lin 36920241153232, Peimeng Wu 36920241153258
AI Class

Abstract

This article aims to improve protein sequence design through deep learning methods. We propose a model that extends the ProteinMPNN framework by introducing a dual layer graph transformer, enabling the capture of multi-scale protein structure information. This model aims to enhance sequence diversity and robustness, and address specific protein design challenges such as protein ligand interactions. We used the Protein Database (PDB) dataset as training data to train the original ProteinMPNN model and our improved model, and compared the training results horizontally, analyzing the possible reasons for this result. Finally, we summarized our topic.

Introduction

The problem of protein sequence design is to find an amino acid sequence that can be folded into a protein skeleton structure of interest given that structure. Rosetta's physics-based approach views sequence design as an energy optimization problem, looking for combinations of amino acid identity and conformation that have the lowest energy for a given input structure. Deep learning methods have shown promise for rapidly generating candidate amino acid sequences given the skeleton of a monomer protein without the need for extensive calculations of side-chain rotational isomerism states. However, the methods described so far are not applicable to the full range of current protein design challenges and have not yet been validated by extensive experiments.

Amino acid sequences at different locations can be coupled between single or multiple chains, enabling applications to a wide range of current protein design tasks. Recent deep learning models are based on the monomeric protein backbone, and they do not need to compute those rotational isomers on the side chain, but their problem is that they are difficult to apply to existing protein design problems, and there is not a lot of experimental validation.

We wanted to address the inefficiencies and inaccuracies encountered with traditional physics-based approaches to protein sequence design. Seeks to improve the speed, accuracy, and applicability of protein design methods to address a wide range of challenges, including the creation of novel protein structures and functions

In order to be able to be applied to a wide range of single - and multi-strand design problems, the fixed N - to C-terminal decoding order is replaced with a sequence-independent autoregressive model, where the decoding order is randomly sampled from the set of all possible permutations, which also leads to modest improvements in sequence recovery. Sequential agnostic decoding can be designed in some cases.

For the multi-chain design problem, in order to make the model sequentially equivalent to the protein chains, the relative position encoding of each chain is kept at ±32 residues and a binary feature is added indicating whether the interacting pair residues are from the same or different chains. The researchers used a flexible decoding sequence to fix the identity of the residues in the corresponding set of locations. For pseudo-symmetric sequence designs, residues within or between chains can be similarly constrained; For example, for repetitive protein designs, the sequence in each repeating unit can remain fixed. By predicting the non-normalized probability of each state and then averaging it, a multi-state design that encodes a single sequence of two or more desired states can be realized.

More generally, a linear combination of the predicted non-normalized probability with some positive and negative coefficients can be used to raise or lower the weight of a particular skeleton state to achieve an unambiguous positive or negative sequence design. The architecture of this multi-chain and symmetric sensing model is called ProteinMPNN.

Related Work

The central challenge in the field of protein design lies in predicting the amino acid sequence of a given protein backbone structure, a problem that is critical for drug discovery, biomaterials development, and other fields. Traditionally, protein design has relied on physically based methods, such as Rosetta, which treat sequence design as an energy optimization problem, searching for the amino acid combination that has the lowest energy for a given input structure. However, these methods are computationally expensive and lack sufficient predictive accuracy, and require extensive explicit consideration of side chain conformational states (Leaver-Fay et al. 2011).

With the development of deep learning techniques, especially in the field of protein structure prediction, new

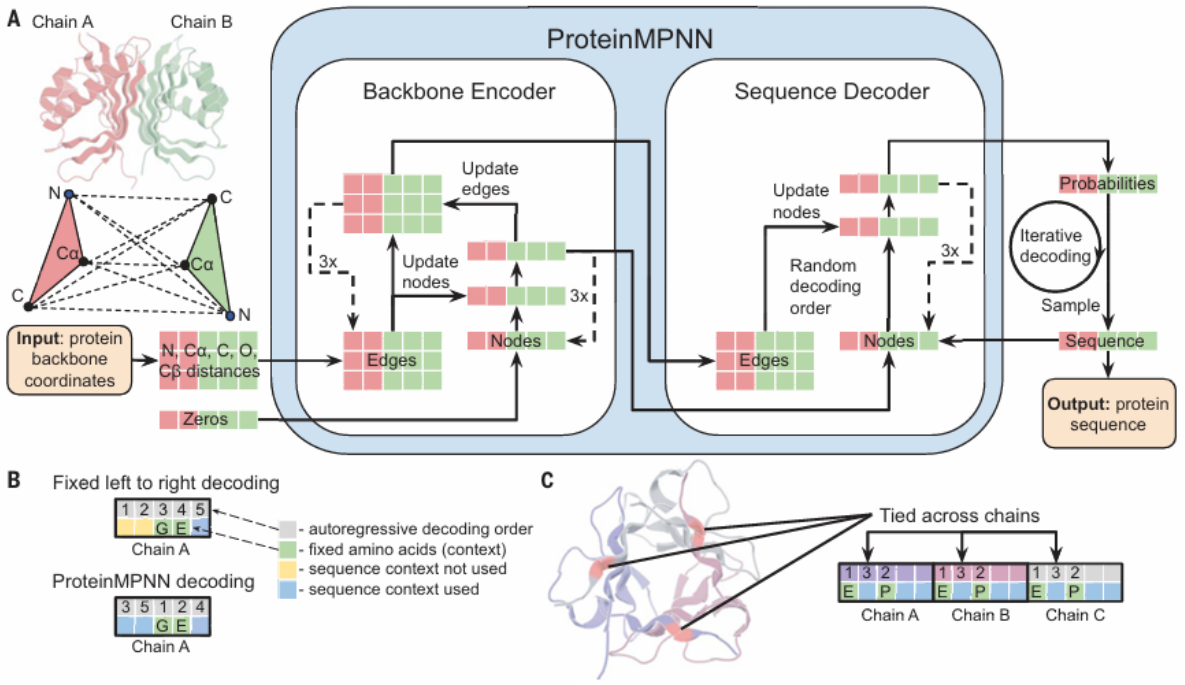

Figure 1: ProteinMPNN architecture. (A) Distances between N,Ca,C,O,and virtual Cβ are encoded and processed using a message-passing neural network(Encoder) to obtain graph node and edge features.The encoded features,together with a partial sequence,are used to generate amino acids iteratively in a random decoding order. (B) A fixed left-to-right decoding cannot use sequence context(green) for preceding positions(yellow), whereas a model trained with random decoding orders can be used with an arbitrary decoding order during the inference.The decoding order can be chosen such that the fixed context is decoded first.(C) Residue positions within and between chains can be tied together,enabling symmetric, repeat protein, and multistate design. In this example,a homotrimer is designed with the coupling of positions in different chains.Predicted unnormalized probabilities for tied positions are averaged to get a single probability distribution from which amino acids are sampled.

learning-based methods are beginning to show their potential. These methods are capable of rapidly generating candidate amino acid sequences without the need for computationally intensive side-chain rotation state considerations (Ingraham et al. 2019; Zhang et al. 2019; Qi and Zhang 2020; Jing et al. 2020; Strokach et al. 2020; Anand et al. 2022; Hsu et al. 2022). The results of these methods emphasize the potential of deep learning in protein design.

(Ingraham et al. 2019) creatively proposed a novel protein design framework based on deep generative modeling and graph representation combined with autoregression and self-attention. The framework is able to effectively handle the complex dependencies between protein sequences and 3D structures, showing better performance. Inspired by this, (Dauparas et al. 2022) proposed the ProteinMPNN model.

ProteinMPNN

ProteinMPNN's structure is shown in figure 1.This model significantly improves the sequence recovery rate by adding additional input features (e.g., distances between N, C, Cα, O, and Cβ) and performing edge updating in a backbone encoder neural network. In addition, ProteinMPNN employs an order agnostic autoregressive model in which the decoding order is randomly sampled from all possible alignments, which allows it to handle the case where the middle part of the protein sequence is fixed. ProteinMPNN performed well in experimental tests, not only surpassing previous methods in sequence recovery, but also showing its utility and accuracy in solving previously failed designs (Jumper et al. 2021; Baek et al. 2021), marking a new milestone in protein sequence design.

Looking ahead, there is still much room for improvement in the application of deep learning to protein design, including improving sequence diversity and robustness, as well as in domain-specific protein sequence design problems, such as protein-ligand interactions and functional design. These approaches offer new possibilities to address a wider range of protein design challenges (Dauparas et al. 2023).So, We attempted to modify the model based on ProteinMPNN.

Proposed Solution

We propose an improved architecture based on the proteinMPNN model. It still follows the encoder-decoder-based message-passing neural network (MPNN) architecture, but the core mechanism utilizes a dual-layer graph Transformer for message passing. By introducing both atomic-level and residue-level dual-layer graph structures, it captures multi-

scale structural information within protein molecules, allowing for finer feature representation and more efficient information propagation.

First, the input module parses the protein's three-dimensional structural data into graph representations that include atoms and residues, and constructs a residue-block-based graph. The dual-layer graph Transformer module then performs message passing and feature updates based on atomic-level and residue-level attention, dynamically modeling the dependencies between atoms and residues. Finally, the output module employs autoregressive decoding to predict the target sequence information using the updated residue features. The key components are described in detail below.

Dual-Layer Graph Transformer

In protein sequence generation tasks, it is crucial to model the complex interactions between residues. However, the multi-granularity nature of these interactions (e.g., atomic-level and residue-level) presents significant challenges. Inspired by the latest research on hierarchical graph transformers, we propose a dual-layer graph transformer to effectively model the multi-granularity interactions in proteins, enabling efficient information transmission and feature updates.

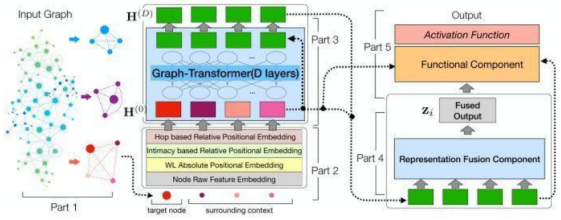

Figure 2: Graph-BERT

Representation Method For ease of representation and computation, we represent each residue as a block. The protein is then abstracted as a geometric graph $G(V, E)$, where $V = \{H_i, X_i \mid 1 \leq i \leq B\}$ represents the set of blocks, and $E = \{e_{ij} \mid 1 \leq i, j \leq B\}$ represents the set of edges. In this representation, $H_i \in R^{n \times d_h}$ represents the atomic features (where n is the number of atoms in the residue, and d_h is the feature dimension), and $X_i \in R^{n \times 3}$ represents the atomic coordinates. Furthermore, $H_i[p]$ and $X_i[p]$ (i.e., the p-th row) represent the learnable features and coordinates of the p-th atom in residue i, respectively.

For the initialization of the learnable features $H_i[p]$, we concatenate embeddings of atomic type, residue, and atomic position. To construct the graph G, we connect residues based on their pairwise C distances, linking the k nearest neighbors. The following section will provide a detailed introduction to the module.

Bilevel Attention Module

The Bilevel Attention Module is designed to capture the interactions at both the atomic and residue levels. First, we assume that block i and block j are connected by edge e_{ij}, and the query (Q), key (K), and value (V) matrices can be obtained through the following linear projection transformations:

$$Q_i = H_i W_Q, \quad K_j = H_j W_K, \quad V_j = H_j W_V \tag{1}$$

where $W_Q, W_K, W_V \in R^{d_h \times d_r}$ are learnable parameters.

Atomic-level Attention To compute the atomic-level attention between block i and block j, we define their relative coordinates and distances as follows:

$$X_{ij}[p, q] = X_i[p] - X_j[q] \tag{2}$$

$$D_{ij}[p, q] = \|X_{ij}[p, q]\|_2 \tag{3}$$

Using the above definitions, we calculate the atomic-level attention coefficients:

$$R_{ij} = \frac{1}{d_r}(Q_i K_j^\top) + \sigma_D(RBF(D_{ij})) \tag{4}$$

$$\alpha_{ij} = Softmax(R_{ij}) \tag{5}$$

where $\sigma_D(\cdot)$ is a multilayer perceptron (MLP) used to add the distance bias into the attention computation, and RBF is the radial basis function embedding. By applying Softmax along the rows of $R_{ij} \in R^{n_i \times n_j}$, we obtain the atomic-level attention matrix $\alpha_{ij} \in R^{n_i \times n_j}$.

Residue-level Attention The residue-level attention from block j to block i is calculated by the following formula:

$$r_{ij} = \frac{\mathbf{1}^\top R_{ij} \mathbf{1}}{n_i n_j} \tag{6}$$

$$\beta_{ij} = \frac{\exp(r_{ij})}{\sum_{j \in \mathcal{N}(i)} \exp(r_{ij})} \tag{7}$$

Here, r_{ij} is the sum of all values in R_{ij}, representing the overall relevance between block i and block j, and $\mathcal{N}(i)$ denotes the set of neighbors of block i. β_{ij} is the residue-level attention coefficient from block j to block i.

Using the atomic-level attention (α_{ij}) and residue-level attention (β_{ij}) described above, we can update the atomic feature representations. For the p-th atom in block i, the feature update is as follows:

$$H_i'[p] = H_i[p] + \sum_{j \in \mathcal{N}(i)} \beta_{ij} \varphi_h(\alpha_{ij}[p] \cdot V_j) \tag{8}$$

Here, φ_h is an MLP, $\alpha_{ij}[p]$ is the attention coefficient for the p-th atom in α_{ij}, and V_j is the value matrix for block j.

Equivariant Feed-Forward Network

We modified the feed-forward network (FFN) module in the graph Transformer model to further update the atomic representations H_i. This update enables the model to capture richer geometric and semantic information within the protein. Specifically, the update of atomic representations incorporates the features of the entire block, and we introduce the concept of a feature centroid.

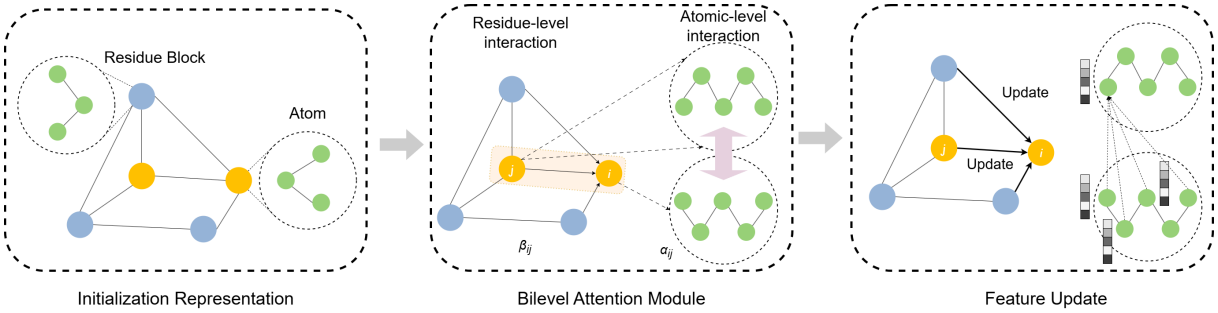

Figure 3: Backbone Encoder Module

The centroid is defined as follows:

$$h_c = centroid(H_i) \tag{9}$$

where $centroid(\cdot)$ denotes the mean of the rows in the matrix (i.e., the average of each atom's features).

Before updating the representation, we first compute the relative coordinates Δx_p of the atom's position and the relative distance representation r_p based on the L2 norm:

$$\Delta x_p = X_i[p] - x_c, \quad r_p = RBF(\|\Delta x_p\|_2) \tag{10}$$

The update of the atomic representation is then defined by the following equations:

$$H_i'[p] = H_i[p] + \sigma_h(H_i[p], h_c, r_p) \tag{11}$$

Here, $\sigma_h(\cdot)$ is an MLP used for the representation update, with input being the concatenation of the atomic feature $H_i[p]$, centroid feature h_c, and relative distance representation r_p.

To stabilize and accelerate training, we apply layer normalization to H in each layer of the Equivariant Dual-Layer Graph Transformer to normalize the features.

Experiments

Dateset

This experiment used biounits from the Protein Data Bank (PDB) as the main data source up to August 2, 2021. PDB is an international resource library that includes information on the structure of biomolecules determined through experimental methods. We chose a complex structure containing two or more polypeptide chains, known as the "multi chain" dataset, to explore and model protein-protein interactions. The multi chain dataset has a total size of approximately 16.5 GB, representing a large-scale and diverse collection of protein structures. Each biounit file contains detailed atomic coordinate information, which is crucial for understanding the spatial conformation of proteins. In addition, to ensure the effectiveness and generalization ability of the model training, we paid special attention to the diversity of the structure in the dataset, including different protein families, functions, and species origins. To evaluate the performance of the model, we divided the dataset into training set, validation set, and test set, with proportions of 70%, 15%, and 15%, respectively. This division helps prevent overfitting and provides a fair environment for comparing the performance of different models. Due to the high computational requirements and limited computing power of this model, only a partial dataset was used for testing in the experiment. All PDB biounits used are publicly available through the PDB official website. For the specific subset of data used in this study, we will provide download links and processing scripts so that other researchers can reproduce our results.PDB Download link.

Evaluation

Sequence Recovery means the percentage of amino acids in the predicted sequence that are consistent with the target sequence.Its calculation formula is as follows

$$SequenceRecovery = \frac{\sum_{i=1}^{N} 1(a_i^{pred} = a_i^{true})}{N} \times 100\% \tag{12}$$

Among them, N represents the total length of the sequence and $1(a_i^{pred} = a_i^{true})$ indicates whether the predicted amino acid a_i^{pred} at the i-th position is consistent with the amino acid a_i^{true} in the real sequence.

Experimental procedure

This project runs on Nvidia V100 and uses the MindSpore deep learning framework. This project can be deployed in different hardware environments by configuring its own operating environment. The environment version used in this project is:

- mindspore-gpu 1.8.0;
- mindspore-ascend 1.9.0;
- python 3.8.

After setting up the environment, we modified the original ProteinMPNN model according to the method mentioned in the Proposed Solution and directly trained it.

Result analysis

The experimental results are shown in Table 1. By comparing the experimental results, it is not difficult to see that the modified model has slightly lower accuracy than the original AlphaFold model. We analyze the possible reasons as follows:

Model Noise level when training: 0.00 Å/0.02 Å	Number of parameters in millions	PDB test acc(%)	PDB test perplexity
ProteinMPNN model	1.381	41.2/40.1	6.51/6.77
Proposed model	1.633	40.2/40.1	6.62/6.64

Table 1: Test accuracy (percentage of correct amino acids recovered) and test perplexity (exponentiated categorical crossentropy loss per residue) for models trained on the native backbone coordinates (value to the left of the slash) and models trained with Gaussian noise (SD = 0.02 A) added to the backbone coordinates (value to the right of the slash).)

1. **Computing power limitations**. Due to the large number of model parameters and our limited computing resources, the number of epochs required for model training may not be sufficient to achieve convergence.
2. **Differences in Model Structure**. ProteinMPNN is a model based on graph neural networks, particularly message passing neural networks. This structure is particularly suitable for processing protein structure like data, as it can effectively capture the interaction information within proteins. Although traditional Transformer models perform well in processing sequential data, they may not be as effective as specialized graph neural networks in handling graph like data.
3. **Feature information loss**. ProteinMPNN considers protein skeleton features such as distance between Cα - Cα atoms, relative Cα - Cα - Cα frame direction and rotation, and backbone dihedral angle as input features during design. These features are crucial for restoring the amino acid sequence of natural single chain proteins. If the dual layer graph Transformer does not effectively utilize these features, it may lead to a decrease in accuracy.

Conclusion

In conclusion, our exploration into enhancing protein sequence design with the advanced deep learning model based on ProteinMPNN has yielded valuable insights and promising results. Our proposed dual-layer graph Transformer model represents a significant advancement in the field of protein design by attempting to integrate multiscale structural information processing within a deep learning framework. Although our model has shown a slight decrease in accuracy compared to the AlphaFold model, this early stage of development provides a foundation for future improvements.

The challenges identified, such as limitations in computing power and the inherent structural differences between graph neural networks and traditional Transformers, are not insurmountable. We anticipate that with increased computational resources, refined model architectures, and further training, our approach can achieve performance on par with or superior to current state-of-the-art methods. The slight performance deficit also suggests areas for potential model enhancement, such as improving the feature extraction process, optimizing the attention mechanisms, and enhancing the model's ability to generalize across diverse protein structures.

Furthermore, our work underscores the importance of continued investment in deep learning for protein design. As the field progresses, we expect deep learning models to play an increasingly pivotal role in the discovery and creation of novel protein structures and functions. The potential applications of such advancements are vast, ranging from drug discovery to the development of new biomaterials, and could ultimately lead to breakthroughs in medicine, agriculture, and environmental science.

In summary, while our proposed model has room for improvement, it represents a step forward in the application of deep learning to protein sequence design. We are encouraged by the initial results and remain committed to the ongoing development of this technology, confident that it will significantly impact the field of protein design and beyond.

References

Anand, N.; Eguchi, R. R.; Mathews, I. I.; Perez, C. P.; Derry, A.; Altman, R. B.; and Huang, P.-S. 2022. Protein sequence design with a learned potential. *Nature Communications*, 13.

Baek, M.; DiMaio, F.; Anishchenko, I.; Dauparas, J.; Ovchinnikov, S.; Lee, G. R.; Wang, J.; Cong, Q.; Kinch, L. N.; Schaeffer, R. D.; Millán, C.; Park, H.; Adams, C.; Glassman, C. R.; DeGiovanni, A.; Pereira, J. H.; Rodrigues, A. V.; van Dijk, A. A.; Ebrecht, A. C.; Opperman, D. J.; Sagmeister, T.; Buhlheller, C.; Pavkov-Keller, T.; Rathinaswamy, M. K.; Dalwadi, U.; Yip, C. K.; Burke, J. E.; Garcia, K. C.; Grishin, N. V.; Adams, P. D.; Read, R. J.; and Baker, D. 2021. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557): 871–876.

Dauparas, J.; Anishchenko, I.; Bennett, N.; Bai, H.; Ragotte, R. J.; Milles, L. F.; Wicky, B. I. M.; Courbet, A.; de Haas, R. J.; Bethel, N.; Leung, P. J. Y.; Huddy, T. F.; Pellock, S.; Tischer, D.; Chan, F.; Koepnick, B.; Nguyen, H.; Kang, A.; Sankaran, B.; Bera, A. K.; King, N. P.; and Baker, D. 2022. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615): 49–56.

Dauparas, J.; Lee, G. R.; Pecoraro, R.; An, L.; Anishchenko, I.; Glasscock, C.; and Baker, D. 2023. Atomic context-conditioned protein sequence design using LigandMPNN. *bioRxiv*.

Hsu, C.; Verkuil, R.; Liu, J.; Lin, Z.; Hie, B.; Sercu, T.; Lerer, A.; and Rives, A. 2022. Learning inverse folding from millions of predicted structures. *bioRxiv*.

Ingraham, J.; Garg, V.; Barzilay, R.; and Jaakkola, T. 2019. Generative Models for Graph-Based Protein Design. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jing, B.; Eismann, S.; Suriana, P.; Townshend, R. J. L.; and Dror, R. O. 2020. Learning from Protein Structure with Geometric Vector Perceptrons. *ArXiv*, abs/2009.01411.

Jumper, J. M.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596: 583 – 589.

Leaver-Fay, A.; Tyka, M.; Lewis, S. M.; Lange, O. F.; Thompson, J.; Jacak, R.; Kaufman, K. W.; Renfrew, P. D.; Smith, C. A.; Sheffler, W.; Davis, I. W.; Cooper, S.; Treuille, A.; Mandell, D. J.; Richter, F.; Ban, Y.-E. A.; Fleishman, S. J.; Corn, J. E.; Kim, D. E.; Lyskov, S.; Berrondo, M.; Mentzer, S.; Popović, Z.; Havranek, J. J.; Karanicolas, J.; Das, R.; Meiler, J.; Kortemme, T.; Gray, J. J.; Kuhlman, B.; Baker, D.; and Bradley, P. 2011. Chapter nineteen - Rosetta3: An Object-Oriented Software Suite for the Simulation and Design of Macromolecules. In Johnson, M. L.; and Brand, L., eds., *Computer Methods, Part C*, volume 487 of *Methods in Enzymology*, 545–574. Academic Press.

Qi, Y.; and Zhang, J. Z. H. 2020. DenseCPD: Improving the Accuracy of Neural-Network-Based Computational Protein Sequence Design with DenseNet. *Journal of chemical information and modeling*.

Strokach, A.; Becerra, D.; Corbi-Verge, C.; Perez-Riba, A.; and Kim, P. M. 2020. Fast and Flexible Protein Design Using Deep Graph Neural Networks. *Cell systems*.

Zhang, Y.; Chen, Y.; Wang, C.; chao Lo, C.; Liu, X.; Wu, W.; and Zhang, J. 2019. ProDCoNN: Protein design using a convolutional neural network. *Proteins: Structure*, 88: 819 – 829.